UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/718,008	11/21/2000	Kenneth Perlin	KPER-4	9323

7590 05/29/2008

Ansel M. Schwartz
One Sterling Plaza
Suite 304
201 N. Craig Street
Pittsburgh, PA 15213

EXAMINER	
WANG, JIN CHENG	

ART UNIT	PAPER NUMBER
2628	

MAIL DATE	DELIVERY MODE
05/29/2008	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	09/718,008	PERLIN, KENNETH	
	Examiner	Art Unit	
	Jin-Cheng Wang	2628	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *07 April 2008*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-12* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-12* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date. _____
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____

DETAILED ACTION

Response to Amendment

Applicant's submission filed on 4/7/2008 has been entered. Claims 1 and 12 have been amended. Claims 1-12 are pending in the application.

Response to Arguments

Applicant's arguments filed April 7, 2008 have been fully considered but are not found persuasive in view of the ground(s) of rejection based on Ebert, D. et al., July 1998, "Texturing and Modeling; A Procedural Approach", Second Edition. AP Professional, Cambridge, pp. 209-274 (hereinafter Ebert et al.).

With respect to the 112 rejection set forth in the prior Office Action, Applicant made general allegation without specifically responding to the rationale of rejection to the claims. For example, applicant speculates "computing a pseudo-random hash value at each vertex of a unit cube C surrounding the point to create a virtual table where entries in a lower half of the table index into a upper half of the table and entries of the upper half of the table index into the lower half of the table" in the claim 1. Applicant refers to the specification at Page 13, lines 15-24 for any teaching of the subject matter. Applicant failed to particularly point out that the specification has any support to the claim limitation set forth above. However, the portion of the specification cited by applicant has no relevance with the specific claim limitation of "computing a pseudo-random hash value at each vertex of a unit cube C surrounding the point to create a virtual table where entries in a lower half of the table index into a upper half of the table and entries of the upper half of the table index into the lower half of the table".

Therefore, Applicant speculates in claim 1 and similar claims the claim limitation of "computing a pseudo-random hash value at each vertex of a unit cube C surrounding the point to create a virtual table where entries in a lower half of the table index into a upper half of the table and entries of the upper half of the table index into the lower half of the table". However, swapping the lower B bits with the upper B bits at the point where the selected data exits from the table does not mean "entries in a lower half of the table index into a upper half of the table and entries of the upper half of the table index into the lower half of the table" as the table has N/2 rows and each row has 2B data bits. It cannot be said "entries in a lower half of the table index into a upper half of the table and entries of the upper half of the table index into the lower half of the table". That there are 2B data bits in each entry in the table is not related to entries in a lower half of the table or entries of the upper half of the table.

Applicant alleges that the cited reference is actually the prior art Perlin noise developed by applicant. The previous version of the Perlin noise was disclosed long time ago by K. Perlin in "An Image Synthesizer", Computer Graphics, Vol. 19, No. 3. See applicant's specification at lines 15-20 of Page 1. The improved Perlin Noise set forth in the cited reference discloses the claimed subject matter. Even though K. Perlin is one of the authors of the cited reference, the cited reference is a printed publication in this country that constitutes a 102(b) reference, which is published more than one year prior to the date of application for patent in the United States.

Applicant argues that various improvements of the claimed invention are listed in regard to Ebert. In contrary to applicant's arguments, applicant claim limitation requires that "so that each 3 dimensional evaluation of one x, y, z triplet requires only one pipelined clock cycle". The cited reference discloses an algorithm in Page 214-218 including the 3-dimensional evaluation of

one x, y, z triplet. This same 3D evaluation is as simple as applicant's claim 1 requires and also requires only one pipelined clock cycle. Since the cited reference teaches the identical functionality, each 3 dimensional evaluation of one x, y, z triplet requires the identical clock time when such evaluation is implemented on the same pipeline. It is also noted that the amended claim 1 also recites the new claim limitation of "resulting in images of computer textures computed in real time". Not surprisingly, the steps of inputting, computing, and combining" set forth in the claim 1 do not require complex computation and thus can be performed in real time, not even to mention the relative computation speed implemented in a old or modern processor. Applicant's claim 1 merely recites each 3D evaluation of one x, y and z triplet requires one clock time. However, such simple 3D evaluation of one x, y and z triplet would require less than one clock time in a modern computer. Computing the textures based on these simple steps, especially the 3D evaluation of one x, y and z triplet, does not require complex computation and thus can be performed in real time. Moreover, Applicant's claim limitation is related the computational speed relative to the speed of a processor for implementing the method. The modern processor runs much faster than the old processor. For example, when the same method is implemented in an old processor (e.g., a 486 Intel Processor), it may take hours or seconds to perform texture computation wherein precisely all steps set forth in the specification are implemented. However, it may take milliseconds to perform texture computation in a modern processor for the same method (e.g., an Intel Pentium II Processor).

It is readily concluded that the execution speed for the same evaluations of the cited prior art versus the claimed subject matter require the same pipelined clock time, relative to the same speed of the microprocessor for implementing the method steps. Moreover, the C-codes

disclosed in Pages 214-218 include each 3 dimensional evaluation of one x, y, z triplet. When each 3 dimensional evaluation of one x, y, z triplet for the prior art is exactly the same as each 3 dimensional evaluation of one x, y, z triplet as claimed, the pipelined clock time for each such evaluation by the cited prior art is equal to the pipelined clock time for each evaluation as claimed. Moreover, it is noted that the C implementation of the cited reference is very efficient, each 3 dimensional evaluation of one x, y, z triplet is so simple that it only requires one clock cycle or maybe less than one block cycle, for example, the C code implemented on the Intel optimizing compiler running on a Pentium 2 or 3 computer.

As addressed below, Ebert et al. including Perlin, the inventor, has disclosed an improved Perlin Noise set forth in applicant's specification. The cited reference discloses a method for creating an appearance of texture in a computer image (see e.g., figures 11-14) comprising the steps of:

Inputting a point (x, y, z) in D-dimensional geometric space R^3 described via six eight-bit quantities i, j, k, u, v, w, wherein i, j, k are the greatest integers not greater than x, y, z, respectively, and u, v, w signify a fractional position of x, y, z above i, j, k to eight-bit precision, in a computer (*Page 213-218 of the cited reference discloses a point [x, y, z], a point [i, j, k] and [u, v, w]*);

Computing a pseudo-random hash value at each vertex of a unit cube C surrounding the point to create a virtual table where entries in a lower half of the table index into a upper half of the table and entries of the upper half of the table index into the lower half of the table (*Page 214-218 of the cited references discloses mapping lattice points [i, j, k] to indices of G, pre-computing a random permutation table P and using this table to fold [i, j, k] into a single n. It*

also discloses computing the gradients G[P[P[P[I]+j]+k]] wherein the precomputed arrays P and G contain a pseudo-random permutation and pseudo-random unit-length gradient vectors wherein the successive application of P hashes each lattice point to de-correlate the indices into G. The eight linear functions G(x-i, y-j, z-k) are then trilinearly interpolated using the cubic approximation, Page 216);

Computing a contribution from each vertex using the hash-value (*Page 214-218 of the cited references discloses mapping lattice points [i, j, k] to indices of G, pre-computing a random permutation table P and using this table to fold [i, j, k] into a single n. It also discloses computing the gradients G[P[P[P[I]+j]+k]] wherein the precomputed arrays P and G contain a pseudo-random permutation and pseudo-random unit-length gradient vectors wherein the successive application of P hashes each lattice point to de-correlate the indices into G. The eight linear functions G(x-i, y-j, z-k) are then trilinearly interpolated using the cubic approximation, Page 216*);

Combining with the computer the contribution from each vertex into a single interpolated result so that each 3 dimensional evaluation of one x, y, z triplet requires only one pipelined clock cycle (*Page 214-218 of the cited references discloses mapping lattice points [i, j, k] to indices of G, pre-computing a random permutation table P and using this table to fold [i, j, k] into a single n. It also discloses computing the gradients G[P[P[P[I]+j]+k]] wherein the precomputed arrays P and G contain a pseudo-random permutation and pseudo-random unit-length gradient vectors wherein the successive application of P hashes each lattice point to de-correlate the indices into G. The eight linear functions G(x-i, y-j, z-k) are then trilinearly interpolated using the cubic approximation, Page 216. The cited reference*

discloses an algorithm in Page 214-218 including the 3-dimensional evaluation of one x, y, z triplet. Since the cited reference teaches the identical functionality, thus each 3 dimensional evaluation of one x, y, z triplet requires the identical CPU time when such evaluation is implemented on the same computer. It is readily concluded that the execution speed for the same evaluations of the prior art versus what is claimed require the same CPU time, e.g., on a computer having the same CPU speed. Moreover, the C-codes disclosed in Pages 214-218 include each 3 dimensional evaluation of one x, y, z triplet and therefore meets the claimed element of "each 3D evaluation of one x, y, z triplet requires only one pipelined clock cycle" as the same 3D evaluation is taught in the cited prior art. When each 3 dimensional evaluation of one x, y, z triplet for the prior art is exactly the same as each 3 dimensional evaluation of one x, y, z triplet as claimed, the CPU time for each such evaluation by the prior art is equal to the CPU time for each evaluation as claimed. Moreover, it is noted that the C implementation of the cited reference is very efficient, each 3 dimensional evaluation of one x, y, z triplet requires only one clock cycle, for example, the C code implemented on the Intel optimizing compiler running on a Pentium 2 or 3 computer) , resulting in images of computer textures computed in real time (*the C-codes disclosed in Pages 214-218 include each 3 dimensional evaluation of one x, y, z triplet and therefore meets the claimed element of "each 3D evaluation of one x, y, z triplet requires only one pipelined clock cycle" as the same 3D evaluation is taught in the cited prior art. See Page 232-240 for computing the computer textures; for the same reasons set forth above, the images of computer textures are computed in real time because the method steps are taught by the cited references*).

Claim Rejections - 35 USC § 112

The following is a quotation of the first paragraph of 35 U.S.C. 112:

> The specification shall contain a written description of the invention, and of the manner and process of making and using it, in such full, clear, concise, and exact terms as to enable any person skilled in the art to which it pertains, or with which it is most nearly connected, to make and use the same and shall set forth the best mode contemplated by the inventor of carrying out his invention.

Claims 1-12 are rejected under 35 U.S.C. 112, first paragraph, as failing to comply with the written description requirement. The claim(s) contains subject matter which was not described in the specification in such a way as to reasonably convey to one skilled in the relevant art that the inventor(s), at the time the application was filed, had possession of the claimed invention.

For example, applicant speculates "computing a pseudo-random hash value at each vertex of a unit cube C surrounding the point to create a virtual table where entries in a lower half of the table index into a upper half of the table and entries of the upper half of the table index into the lower half of the table" in the claim 1.

From the applicant's specification, Pages 14-15 and Pages 25-31, pseudo-random hash values are computed. It is stated, "As in Perlin, K., An Image Synthesizer, Computer Graphics; Vol. 19 No. 3, incorporated by reference herein, this computation steps through the coordinates, doing alternating lookups and adds: L(L(L(i) + j) + k)), where function L does a table look-up of its argument, modulo 128, into a pseudo-random table of stored values. This alternation of lookups and offsets into a pseudo-random table prevents correlations between the values returned at neighboring locations on the integer coordinate grid, which would otherwise appear as unwanted visible patterns."

"Since there are 8 vertices, and three lookups are required per vertex, this would appear to require 24 table lookups, which would be quite expensive in the number of gates required. This requirement is reduced by implementing L as a lookup table which simultaneously retrieves two successive table entries."

"The table is implemented as follows: Instead of an N row table with B data bits per row (in the current embodiment, N = 128 and B = 7 or 5), a table is laid out which has N/2 rows with 2B data bits per row. The top B-I control bits are used to select a row r in the standard manner for a ROM implementation of a lookup table. The lowest control bit does two things: (i) Latch between selecting entry r (when the lowest control bit is clear) and r+l (when the lowest control bit is set) for the lowest B bits. (ii) While the lowest control bit is set, swap the lower B bits with the upper B bits at the point where the selected data exits from the table."

"The method disclosed requires somewhat more gates per bit of storage than is required for a simple N x B table, but far fewer than would be required to maintain two independent N x B tables."

"As shown in figure 1, i is fed into the first L module, which produces a result for both i and i+l. Then each of these results is added to j and fed into two L modules, which produces results for (i, j), (i+l, j), (i, j+l) and (i+l, j+l). Finally, these results are added to k, and fed into four L modules, thereby producing the required eight hash values."

Nothing in the applicant's specification can be used to support the new claim limitation "computing a pseudo-random hash value at each vertex of a unit cube C surrounding the point to create a virtual table where entries in a lower half of the table index into a upper half of the table

and entries of the upper half of the table index into the lower half of the table" set forth in the claim 1.

Or example, swapping the lower B bits with the upper B bits at the point where the selected data exits from the table does not mean "entries in a lower half of the table index into a upper half of the table and entries of the upper half of the table index into the lower half of the table" as the table has N/2 rows and each row has 2B data bits. It cannot be said "entries in a lower half of the table index into a upper half of the table and entries of the upper half of the table index into the lower half of the table". 2B data bits in each entry in the table has nothing to do with entries in a lower half of the table or entries of the upper half of the table.

Finally, applicant's specification failed to support the claim limitation "each 3 dimensional evaluation of one x, y, z triplet requires only one pipelined clock cycle". This is because the triplets (x, y, z) constitute an input point. Since the input point is already known, it does not make sense that an input is evaluated and each 3 dimensional evaluation of an input point requires only one pipelined clock cycle. See Page 12 of applicant's specification wherein the evaluation refers to the whole process for computing a noise value including the steps of hashing, gradient and interpolation, as opposed to solely evaluating the triplets (x, y, z). Page 12 only discloses accepting the triplets (x, y, z) per clock cycle. This does not mean evaluation of the triplets (x, y, z) requires a clock cycle.

Claims 2-11 depend upon the claim 1 and are rejected due to their dependency on the claim 1.

The claim 12 is subject to the same rationale of rejection set forth in the claim 1.

Claim Rejections - 35 USC § 112

The following is a quotation of the second paragraph of 35 U.S.C. 112:

> The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

Claims 1-12 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

For example, the base claim 1 recites "computing a pseudo-random hash value at each vertex of a unit cube C surrounding the point to create a virtual table where entries in a lower half of the table index into a upper half of the table and entries of the upper half of the table index into the lower half of the table".

Swapping the lower B bits with the upper B bits at the point where the selected data exits from the table does not mean "entries in a lower half of the table index into a upper half of the table and entries of the upper half of the table index into the lower half of the table" as the table has N/2 rows and each row has 2B data bits. It cannot be said "entries in a lower half of the table index into a upper half of the table and entries of the upper half of the table index into the lower half of the table". 2B data bits in each entry in the table has nothing to do with entries in a lower half of the table or entries of the upper half of the table. Clarification is required.

Claims 2-11 depend upon the claim 1 and are rejected due to their dependency on the claim 1.

The claim 12 is subject to the same rationale of rejection set forth in the claim 1.

Claim Rejections - 35 USC § 102

The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

1. Claims 1 and 12 are rejected under 35 U.S.C. 102(b) as being anticipated by Ebert, D. et al., July 1998, "Texturing and Modeling; A Procedural Approach", Second Edition. AP Professional, Cambridge, pp. 209-274 (hereinafter Ebert et al.).

2. Re Claims 1 and 12:

Ebert et al. including Perlin, the inventor, has disclosed an improved Perlin Noise set forth in applicant's specification. The cited reference discloses a method for creating an appearance of texture in a computer image (see e.g., figures 11-14) comprising the steps of:

Inputting a point (x, y, z) in D-dimensional geometric space R^3 described via six eight-bit quantities i, j, k, u, v, w, wherein i, j, k are the greatest integers not greater than x, y, z, respectively, and u, v, w signify a fractional position of x, y, z above I, j, k to eight-bit precision, in a computer (*Page 213-218 of the cited reference discloses a point [x, y, z], a point [i, j, k] and [u, v, w]*);

Computing a pseudo-random hash value at each vertex of a unit cube C surrounding the point to create a virtual table where entries in a lower half of the table index into a upper half of

the table and entries of the upper half of the table index into the lower half of the table (*Page 214-218 of the cited references discloses mapping lattice points [i, j, k] to indices of G, pre-computing a random permutation table P and using this table to fold [i, j, k] into a single n. It also discloses computing the gradients G[P[P[P[I]+j]+k]] wherein the precomputed arrays P and G contain a pseudo-random permutation and pseudo-random unit-length gradient vectors wherein the successive application of P hashes each lattice point to de-correlate the indices into G. The eight linear functions G(x-i, y-j, z-k) are then trilinearly interpolated using the cubic approximation, Page 216*);

Computing a contribution from each vertex using the hash-value (*Page 214-218 of the cited references discloses mapping lattice points [i, j, k] to indices of G, pre-computing a random permutation table P and using this table to fold [i, j, k] into a single n. It also discloses computing the gradients G[P[P[P[I]+j]+k]] wherein the precomputed arrays P and G contain a pseudo-random permutation and pseudo-random unit-length gradient vectors wherein the successive application of P hashes each lattice point to de-correlate the indices into G. The eight linear functions G(x-i, y-j, z-k) are then trilinearly interpolated using the cubic approximation, Page 216*);

Combining with the computer the contribution from each vertex into a single interpolated result so that each 3 dimensional evaluation of one x, y, z triplet requires only one pipelined clock cycle (*Page 214-218 of the cited references discloses mapping lattice points [i, j, k] to indices of G, pre-computing a random permutation table P and using this table to fold [i, j, k] into a single n. It also discloses computing the gradients G[P[P[P[I]+j]+k]] wherein the precomputed arrays P and G contain a pseudo-random permutation and pseudo-random*

unit-length gradient vectors wherein the successive application of P hashes each lattice point to de-correlate the indices into G. The eight linear functions G(x-i, y-j, z-k) are then trilinearly interpolated using the cubic approximation, Page 216. The cited reference discloses an algorithm in Page 214-218 including the 3-dimensional evaluation of one x, y, z triplet. Since the cited reference teaches the identical functionality, thus each 3 dimensional evaluation of one x, y, z triplet requires the identical CPU time when such evaluation is implemented on the same computer. It is readily concluded that the execution speed for the same evaluations of the prior art versus what is claimed require the same CPU time, e.g., on a computer having the same CPU speed. Moreover, the C-codes disclosed in Pages 214-218 include each 3 dimensional evaluation of one x, y, z triplet and therefore meets the claimed element of "each 3D evaluation of one x, y, z triplet requires only one pipelined clock cycle" as the same 3D evaluation is taught in the cited prior art. When each 3 dimensional evaluation of one x, y, z triplet for the prior art is exactly the same as each 3 dimensional evaluation of one x, y, z triplet as claimed, the CPU time for each such evaluation by the prior art is equal to the CPU time for each evaluation as claimed. Moreover, it is noted that the C implementation of the cited reference is very efficient, each 3 dimensional evaluation of one x, y, z triplet requires only one clock cycle, for example, the C code implemented on the Intel optimizing compiler running on a Pentium 2 or 3 computer) , resulting in images of computer textures computed in real time (*the C-codes disclosed in Pages 214-218 include each 3 dimensional evaluation of one x, y, z triplet and therefore meets the claimed element of "each 3D evaluation of one x, y, z triplet requires only one pipelined clock cycle" as the same 3D evaluation is taught in the cited prior art. See Page 232-240 for computing the*

computer textures; for the same reasons set forth above, the images of computer textures are computed in real time because the method steps are taught by the cited references).

Claim Rejections - 35 USC § 103

3. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

4. Claims 2-11 are rejected under 35 U.S.C. 103(a) as being unpatentable over Ebert, D. et al., July 1998, "Texturing and Modeling; A Procedural Approach", Second Edition. AP Professional, Cambridge, pp. 209-274 (hereinafter Ebert et al.) in view of David M. Lewis "Procedural Texture Mapping on FPGAs", ACM 1999, 1-58113-088-0/99/02, page 112-120 (hereinafter Ye).

5. Claims 2-4:

Ebert et al teach combining the contribution from each vertex into a single result using 3 ease-curve s modules (Page 216).

(2) Ebert et al do not teach (a) six "+" modules combined with seven "L" modules; (b) three "+" modules combined with eight "H" modules; (c) the s modules.

(3) Ye however teaches (a) the "+" modules (figure 8); (b) the "L" modules and the "H" modules (figures 7 and 8); (c) the s modules (figures 6 and 10).

(4) It would have been obvious to one of ordinary skill in the art to have incorporated the various combinations of "+" modules, the "L" modules, "H" modules and the s modules into the Ebert et al.'s method for creating a noise because Ye suggests implementing "+" modules in figure 8, XOR modules in figure 8 and s modules in figures 6 and 10 and therefore suggesting an obvious modification.

(5) Therefore, it would have been obvious to implement Ebert's method with some specific numbers/combinations of modules so that it would facilitate an efficient implementation of Perlin Noise based 3-D procedural textures.

Claim 5:

The claim 5 encompasses the same scope of invention as that of claim 4 except additional claimed limitation of a look-up table. However, Ye further discloses the claimed limitation of a look-up table (Ye page 116) and Ebert et al teach the lookup table (Ebert Page 209-274).

Claim 6:

The claim 6 encompasses the same scope of invention as that of claim 5 except additional claimed limitation of computing a gradient direction from each hash value. However, Ye and Ebert further disclose the claimed limitation of computing a gradient direction from each hash value (Ye page 116 and Ebert Pages 209-274).

Claim 7:

The claim 7 encompasses the same scope of invention as that of claim 6 except additional claimed limitation of allowing the inner product to be done using no multiples, only adds and shifts. However, Ye and Ebert further disclose the claimed limitation of allowing the inner

product to be done using no multiples, only adds and shifts (Ye figures 6-10 and Ebert Page 209-274).

Claim 8:

The claim 8 encompasses the same scope of invention as that of claim 7 except additional claimed limitation of choosing the gradients. However, Ye and Ebert further disclose the claimed limitation of choosing the gradients (Ye page 116 and Ebert Page 209-274).

Claim 9:

The claim 9 encompasses the same scope of invention as that of claim 8 except additional claimed limitation of using 7 linear-interrelation modules L to perform a trilinear interpolation. However, Ye and Ebert further disclose the claimed limitation of using 7 linear-interrelation modules L to perform a trilinear interpolation (Ye pages 115-116 and Ebert Page 209-274).

Claim 10:

The claim 10 encompasses the same scope of invention as that of claim 9 except additional claimed limitation of computing an ease curve. However, Ye and Ebert further disclose the claimed limitation of computing an ease curve (Ye page 116 and Ebert Page 209-274).

Claim 11:

The claim 11 encompasses the same scope of invention as that of claim 10 except additional claimed limitation of linear interpolations modules. However, Ye and Ebert further disclose the claimed limitation of linear interpolations modules (Ye figure 7 and Ebert Page 209-274).

Conclusion

Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Jin-Cheng Wang whose telephone number is (571) 272-7665. The examiner can normally be reached on 8:00 - 6:30 (Mon-Thu).

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Richard Hjerpe can be reached on (571) 272-7691. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

/Jin-Cheng Wang/
Primary Examiner, Art Unit 2628